

### **REMARKS**

Applicant appreciates the time taken by the Examiner to review Applicant's present application. This application has been carefully reviewed in light of the Official Action mailed June 30, 2008. Applicant respectfully requests reconsideration and favorable action in this case.

### **CLAIMS STATUS**

Claims 1-53 were pending and rejected. Claims 1, 2, 4, 10, 14, 16, 17, 18, 19, 23, 25, 29, 31, 32, 34, 40 and 46 are amended herein. Claims 6, 7, 9, 21, 22, 24, 36, 37, 39 and 51-53 are cancelled. No new claims are added. Support for the amendments presented herein can be found in the specification as originally filed. See e.g., Specification, paras. 51-59, 72, and 80-83. No new matter is introduced. By this Amendment, claims 1-5, 8, 10-20, 23, 25-35, 38 and 40-50 are pending.

### **Rejections under 35 U.S.C. § 112**

Claims 21-30 and 40-45 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 21 is cancelled herein. Claims 23, 25 and 40 are amended herein. Accordingly, withdrawal of this rejection is respectfully requested.

### **Rejections under 35 U.S.C. § 101**

Claims 31-39 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Claim 31 is amended herein. Accordingly, withdrawal of this rejection is respectfully requested.

**Rejections under 35 U.S.C. § 103**

Claims 1-53 were rejected under 35 U.S.C. §103(a) as being unpatentable over “*Database Design for Smarties Using UML for Data Modeling*” by Robert Muller (“Muller”). The rejection is respectfully traversed. Claims 16 and 31 contain similar language as claim 1. Therefore, the rejection will be addressed collectively as it pertains to claim 1.

Claim 1, as amended, recites:

A method of modeling an arbitrarily complex environment, comprising:  
on a computer having a computer memory and a processor, defining a plurality of types of  
data structures in a data model, wherein each of the data structures comprises  
one or more fields or properties associated with the data structure, wherein all  
data structures of the same type contain the same properties;  
instantiating a component for each atomic entity in the arbitrarily complex environment,  
wherein each component has a set of fields which contain information relating to  
the atomic entity associated with the component, wherein the set of fields  
comprises:  
a set of property fields containing information about the attributes or  
characteristics of the component; and  
a field that contains a link to its component type;  
assigning values to the properties in the instantiated component based on the attributes of  
the entity which the component was instantiated to represent;  
instantiating a relationship for representing an association or a dependency between two  
or more components in the data model, wherein each relationship comprises:  
a field that is a foreign key to its relationship type; and  
a set of property fields containing information about one or more of the  
attributes of the relationship; and  
storing the components in a schema, wherein property definitions of each component are  
linked to a type of component, wherein changes made to the type of component  
are automatically associated with all components of that type of component  
without changing the schema to reflect a corresponding change in the arbitrarily  
complex environment.

Thus, embodiments of a method for modeling an arbitrarily complex environment disclosed by Applicant may include defining a plurality of types of data structures in a data model, wherein each of the data structures comprises one or more fields of properties

associated with the data structure, wherein all data structures of the same type contain the same properties. Using the defined data structures, instances of types of data structures may be used to represent entities in the environment as components and associations between components as relationships. Each component has a set of fields including a property field containing information about the attributes or characteristics of the component and a field that links the component to its component type. Each relationship has a set of fields including a property field containing information about one or more of the attributes of the relationship and a field that links the relationship to its relationship type. The components and relationships may be stored in a schema such that property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with each component of that type of component without changing the schema, and wherein one or more fields in a relationship are changed to reflect a change in the arbitrarily complex environment.

Thus, in addition to providing a generic data model for modeling an arbitrarily complex environment and creating a schema, embodiments may reflect changes to the environment without rewriting significant amounts of code. For example, a server computer may have a set of fields which are generic to all server computers. The fields in the component representing the server component may have selected values to associate the component with a particular server computer, associate the component with a particular type of component, contain properties based on the type of component, and may be assigned values corresponding with the particular server. If any of the information must be changed to reflect a change in the arbitrarily complex environment, only changes to the information contained in the fields needs to be changed – the entire component does not need to be re-coded. (See, specification, paras. 51-58.)

Similarly, a relationship may be built programmatically to reflect changes in the arbitrarily complex environment. The property fields may allow attributes of the relationships to be represented by a name and value pair. In one particular example, a first field may be set to a name corresponding to a first component associated by the relationship and a second field may be set to a name corresponding to a second component. The relationship may have a field that links the relationship to a relationship type. The data model may be stored in a schema, in for example, a table. If either component is renamed, the relationship does not need to be altered (i.e, re-coded), and the arbitrarily complex environment is still modeled, by, for example, changing the name in the table. (See, specification, paras. 59-61).

Applicant respectfully submits that Muller generally describes the prior art entity-relationship modeling of an environment in a way such that the methods and systems employed by Muller may be used to model an arbitrarily complex environment, but the model would be hard-coded and any changes to the arbitrarily complex environment would be time-consuming to change in the schema. In fact, Applicant is unable to find an example in Muller in which the object could be changed without changing the code for that object. Applicant submits that disadvantages to the prior art taught by Muller include manual creation of documents, whole documentation systems must be kept in place to version and store documents associated with the models, the model being prone to errors, a large number of users may need access to the model concurrently, cross-referencing information across the model is difficult, and the models are susceptible to changes in the environment. (See, specification, paras. 8-9). For at least the foregoing reasons, Applicant respectfully submits that the teachings of Muller are concerned with prior art approaches to modeling entity-relationship models, and fail to teach or describe one or more of: defining a plurality of types of data structures in a data model, wherein each of the data structures comprises one or more fields or properties associated with the data structure, wherein all data structures of the same type contain the same properties; instantiating a component for each atomic entity in the arbitrarily complex environment, wherein each component has a set of fields which contain information relating to the atomic entity associated with the component, wherein the set of fields comprises a set of property fields containing information about the attributes or characteristics of the component and a field that is a foreign key to its component type; assigning values to the properties in the instantiated component based on the attributes of the entity which the component was instantiated to represent; instantiating a relationship for representing an association or a dependency between two or more components in the data model, wherein each relationship comprises a field that is a foreign key to its relationship type; and a set of property fields containing information about one or more of the attributes of the relationship; and storing the components in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema to reflect a corresponding change in the arbitrarily complex environment, as recited in Claim 1. Accordingly, withdrawal of this rejection is requested.

Conclusion

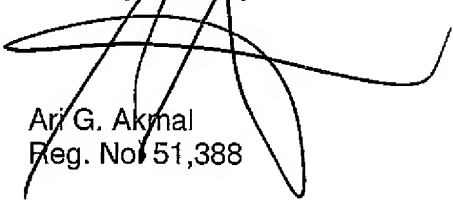
Applicant has now made an earnest attempt to place this case in condition for allowance. Other than as explicitly set forth above, this reply does not include an acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, Applicant respectfully requests full allowance of Claims 1-5, 8, 10-20, 23, 25-35, 38 and 40-50. The Examiner is invited to telephone the undersigned at the number listed below for prompt action in the event any issues remain.

An extension of 2 (two) months is requested and a Notification of Extension of Time Under 37 C.F.R. § 1.136 with the appropriate fee is enclosed herewith.

The Director of the U.S. Patent and Trademark Office is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

**Sprinkle IP Law Group**  
Attorneys for Applicant



Ari G. Akmal  
Reg. No. 51,388

Date: December 1, 2008

1301 W. 25<sup>th</sup> Street, Suite 408  
Austin, TX 78705  
Tel. (512) 637-9220  
Fax. (512) 371-9088